WEST

Help Logout Interrupt

Main Menu | Search Form | Posting Counts | Show S Numbers | Edit S Numbers | Preferences | Cases

Search Results -

Terms	Documents
L5 AND ((717/$$$)!.CCLS.)	18

Database:

US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L6

Refine Search

Recall Text Clear

Search History

DATE: Sunday, December 14, 2003 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=NO; OP=OR			
L6	L5 AND ((717/$$$)!.CCLS.)	18	L6
L5	L4 AND object	81	L5
L4	L2 and (attribute or variable)	82	L4
L3	L2 and (attribute or variable)	82	L3
L2	L1 AND default	85	L2
L1	getter and setter	117	L1

END OF SEARCH HISTORY

WEST

Help | Logout | Interrupt

Main Menu | Search Form | Posting Counts | Show S Numbers | Edit S Numbers | Preferences | Cases

Search Results -

Terms	Documents
L1 AND ((717/$$$)!.CCLS.)	1

Database:
US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L2

Refine Search

Recall Text | Clear

Search History

DATE: Sunday, December 14, 2003 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=NO; OP=OR		
L2	L1 AND ((717/$$$)!.CCLS.)	1	L2
L1	(4931974 6125334 4559611 4953101 5287520 5465375 5487173 5491828 5511219 5519879 5553135 5590357 5592677 5596764 5603017 5606714 5613149 5625828 5638306 5649208 5671372 5867382 5872960 5943341 6002814 6034542 6076142 6122619 6128597 6154789 6243767 6253193 6307659 6363488 6389402 6394263 6427140 5367653 5748875 5845317 6023564 6105105 6148374 6223274 6223274 4845706 5014195 5253940 5469566 5621723).pn.	49	L1

END OF SEARCH HISTORY

WEST

Generate Collection | Print

Search Results - Record(s) 1 through 1 of 1 returned.

☑ 1. Document ID: US 6223274 B1

L2: Entry 1 of 1 File: USPT Apr 24, 2001

US-PAT-NO: 6223274
DOCUMENT-IDENTIFIER: US 6223274 B1

TITLE: Power-and speed-efficient data storage/transfer architecture models and design methodologies for programmable or reusable multi-media processors

DATE-ISSUED: April 24, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Catthoor; Francky	Temse			BE
Miranda; Miguel	Heverlee			BE
Janssens; Stefan	Lennik			BE
De Man; Hugo	Kessel-Lo			BE

US-CL-CURRENT: 712/34; 709/108, 712/40, 717/127

ABSTRACT:

A programmable processing engine and a method of operating the same is described, the processing engine including a customized processor, a flexible processor and a data store commonly sharable between the two processors. The customized processor normally executes a sequence of a plurality of pre-customized routines, usually for which it has been optimized. To provide some flexibility for design changes and optimizations, a controller for monitoring the customized processor during execution of routines is provided to select one of a set of pre-customized processing interruption points and for switching context from the customized processor to the flexible processor at the interruption point. The customized processor can then be switched off and the flexible processor carries out a modified routine. By using sharable a data store, the context switch can be chosen at a time when all relevant data is in the sharable data store. This means that the flexible processor can pick up the modified processing cleanly. After the modified processing the flexible processor writes back new data into the data store and the customized processor can continue processing either where it left off or may skip a certain number of cycles as instructed by the flexible processor, before beginning processing of the new data.

30 Claims, 27 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 23

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

Generate Collection | Print

Terms	Documents
L1 AND ((717/$$$)!.CCLS.)	1

Display Format: REV Change Format

Previous Page Next Page

WEST

Help | Logout | Interrupt

Main Menu | Search Form | Posting Counts | Show S Numbers | Edit S Numbers | Preferences | Cases

Search Results -

Terms	Documents
L5 AND ((717/$$$)!.CCLS.)	18

Database:

- US Patents Full-Text Database
- US Pre-Grant Publication Full-Text Database
- JPO Abstracts Database
- EPO Abstracts Database
- Derwent World Patents Index
- IBM Technical Disclosure Bulletins

Search: L6

Refine Search

Recall Text | Clear

Search History

DATE: Sunday, December 14, 2003 Printable Copy Create Case

Set Name **side by side**	**Query**	**Hit Count**	**Set Name** **result set**
	DB=USPT; PLUR=NO; OP=OR		
L6	L5 AND ((717/$$$)!.CCLS.)	18	L6
L5	L4 AND object	81	L5
L4	L2 and (attribute or variable)	82	L4
L3	L2 and (attribute or variable)	82	L3
L2	L1 AND default	85	L2
L1	getter and setter	117	L1

END OF SEARCH HISTORY

WEST

Generate Collection | Print

Search Results - Record(s) 1 through 18 of 18 returned.

☑ 1. Document ID: US 6654932 B1

L6: Entry 1 of 18 File: USPT Nov 25, 2003

US-PAT-NO: 6654932
DOCUMENT-IDENTIFIER: US 6654932 B1

TITLE: Validating data within container objects handled by view controllers

DATE-ISSUED: November 25, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bahrs; Peter C.	Austin	TX		
Modh; Manish Mahesh	Round Rock	TX		

US-CL-CURRENT: 715/507; 345/764, 715/508, 717/116

ABSTRACT:

A method and apparatus in a data processing system for performing validation of user input. User input is received in a container displayed in a graphical user interface, wherein presentation of the container and the user input to the container are handled by a view controller. Responsive to receiving the user input, a call is sent to a validation object by the view controller. Responsive to the call, the validation object tests the user input using a criteria, wherein the rule is separate from the view controller.

25 Claims, 196 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 119

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

☑ 2. Document ID: US 6654778 B1

L6: Entry 2 of 18 File: USPT Nov 25, 2003

US-PAT-NO: 6654778
DOCUMENT-IDENTIFIER: US 6654778 B1

TITLE: Method and apparatus for avoiding function activation and interpretation overhead for calls to selected java methods in a java virtual machine interpreter

DATE-ISSUED: November 25, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Blandy; Geoffrey Owen	Austin	TX		
Hargrave; Bentley John	Austin	TX		

US-CL-CURRENT: 718/1; 717/118, 717/136, 717/148

ABSTRACT:

A method and apparatus for a process in a computer for processing a method that performs a function. A determination is made as to whether the method is to be executed normally when the method is loaded. Responsive to an absence of a determination that the method is a method to be executed normally, instructions native to the computer are associated with the method, wherein the instructions perform the function.

26 Claims, 7 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 3

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

☐ 3. Document ID: US 6601234 B1

L6: Entry 3 of 18 File: USPT Jul 29, 2003

US-PAT-NO: 6601234
DOCUMENT-IDENTIFIER: US 6601234 B1

TITLE: Attribute dictionary in a business logic services environment

DATE-ISSUED: July 29, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bowman-Amuah; Michel K.	Colorado Springs	CO		

US-CL-CURRENT: 717/108; 705/7, 717/107, 717/116

ABSTRACT:

A system and method are provided for controlling access to data of a business object via an attribute dictionary. The attribute dictionary, which stores attribute names and values, is dispatched over a network. A helper facade is provided for interfacing a business object and the attribute dictionary. Next, it is verified that a current user is authorized to either set or get one of the attribute values upon a request which includes the attribute name that corresponds to the attribute value. The helper facade is called to set, get, or update one of the attribute values based on the corresponding attribute name, wherein the helper facade shields the attribute dictionary from the application code of the business object. The attribute value in the attribute dictionary is obtained or updated if the verification is successful, and a dirty flag is set in the attribute dictionary and an indicator is broadcast upon the attribute value being updated.

15 Claims, 195 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 123

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

☐ 4. Document ID: US 6598219 B1

L6: Entry 4 of 18 File: USPT Jul 22, 2003

US-PAT-NO: 6598219

DOCUMENT-IDENTIFIER: US 6598219 B1

TITLE: Method and mechanism for a task oriented XML data model

DATE-ISSUED: July 22, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Lau; Christina P.	Scarborough			CA

US-CL-CURRENT: 717/108

ABSTRACT:

A task oriented data model for an object oriented development tool. The data model comprises a task oriented structure which mirrors the task tree embodied in the user interface of the object development tool. The object development tool exports the data model as a document expressed in meta data language such as XML. The XML data model document comprises a series of data elements which are arranged according to a meta data model to mirror the user interface task tree. The meta data model is implemented in XML as a Document Type Definition and comprises containment relationships defined by XML constructs. The task oriented data model provides granularity for exporting and importing data elements, and also facilitates locating data elements of interest in the data model.

12 Claims, 2 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 2

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments   KWIC | Draw Desc | Image

---

□ 5. Document ID: US 6574791 B1

L6: Entry 5 of 18   File: USPT   Jun 3, 2003

US-PAT-NO: 6574791
DOCUMENT-IDENTIFIER: US 6574791 B1

TITLE: Component based designer for modifying and specializing wizards

DATE-ISSUED: June 3, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Gauthier; Charles Salo	Stewartville	MN		
Minard; Kevin Glen	Parker	CO		
Toogood; Jay Daniel	Rochester	MN		

US-CL-CURRENT: 717/107; 717/109

ABSTRACT:

The preferred embodiment of the present invention an object oriented wizard creation mechanism provides an environment and set of facilities for creating and modifying wizards. The preferred embodiment wizard creation mechanism provides a WizardWizard mechanism for creating the skeleton of new wizard, a WizardDesigner mechanism for specializing and WizardMetaDataManager mechanism for persisting and retrieving created wizards. The WizardWizard guides the developer through a predetermined series of steps that are required to define a the basic components of a new wizard. The WizardDesigner takes these basic components and guides the developer through a specialization process that further customizes and defines the new wizard. In the preferred embodiment, the WizardWizard and the WizardDesigner create and specialize the target wizard from a component based Wizard framework. The use of the component based Wizard framework with the WizardWizard and WizardDesigner provide a customizable and extensible wizard creation solution to that has utmost functionality and flexibility to the users and developers.

63 Claims, 28 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 28

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments   KWIC | Draw Desc | Image

☑ 6. Document ID: US 6550057 B1

L6: Entry 6 of 18   File: USPT   Apr 15, 2003

US-PAT-NO: 6550057
DOCUMENT-IDENTIFIER: US 6550057 B1

TITLE: Piecemeal retrieval in an information services patterns environment

DATE-ISSUED: April 15, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bowman-Amuah; Michel K.	Colorado Springs	CO		

US-CL-CURRENT: 717/126; 700/80, 707/5, 717/101, 717/102, 717/108, 717/109, 717/113

ABSTRACT:

A system, method and article of manufacture are provided for providing a warning upon retrieval of objects that are incomplete. An object is provided with at least one missing attribute. Upon receipt of a request from an application for the object access to the attributes of the object is allowed by the application. A warning is provided upon an attempt to access the attribute of the object that is missing.

15 Claims, 195 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 123

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments   KWIC | Draw Desc | Image

□ 7. Document ID: US 6502234 B1

L6: Entry 7 of 18   File: USPT   Dec 31, 2002

US-PAT-NO: 6502234
DOCUMENT-IDENTIFIER: US 6502234 B1

TITLE: Component based wizard for creating wizards

DATE-ISSUED: December 31, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Gauthier; Charles Salo	Stewartville	MN		
Minard; Kevin Glen	Parker	CO		
Toogood; Jay Daniel	Rochester	MN		

US-CL-CURRENT: 717/107; 717/101, 717/106, 717/108, 717/109, 717/115, 717/116

ABSTRACT:

The preferred embodiment of the present invention an object oriented wizard creation mechanism provides an environment and set of facilities for creating and modifying wizards. The preferred embodiment wizard creation mechanism provides a WizardWizard mechanism for creating the skeleton of new wizard, a WizardDesigner mechanism for specializing and WizardMetaDataManager mechanism for persisting and retrieving created wizards. The WizardWizard guides the developer through a predetermined series of steps that are required to define a the basic components of a new wizard. The WizardDesigner takes these basic components and guides the developer through a specialization process that further customizes and defines the new wizard. In the preferred embodiment, the WizardWizard and the WizardDesigner create and specialize the target wizard from a component based Wizard framework. The use of the component based Wizard framework with the WizardWizard and WizardDesigner provide a customizable and extensible wizard creation solution to that has utmost functionality and flexibility to the users and developers.

54 Claims, 28 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 28

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments   KWIC | Draw Desc | Image

☑ 8. Document ID: US 6442748 B1

L6: Entry 8 of 18 File: USPT Aug 27, 2002

US-PAT-NO: 6442748
DOCUMENT-IDENTIFIER: US 6442748 B1

TITLE: System, method and article of manufacture for a persistent state and persistent object separator in an information services patterns environment

DATE-ISSUED: August 27, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bowman-Amuah; Michel K.	Colorado Springs	CO		

US-CL-CURRENT: 717/108; 707/103R, 707/104.1, 707/9, 709/316

ABSTRACT:

A system, method and article of manufacture are provided for separating logic and data access concerns during development of a persistent object for insulating development of business logic from development of data access routine. A persistent object being developed is accessed and a state of the persistent object is detached into a separate state class. The state class serves as a contract between a logic development team and a data access development team. Logic development is limited by the logic development team to developing business logic. Data access development is restricted by the data access development team to providing data creation, retrieval, updating, and deletion capabilities.

18 Claims, 195 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 123

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments   KWIC | Draw Desc | Image

□ 9. Document ID: US 6314559 B1

L6: Entry 9 of 18 File: USPT Nov 6, 2001

US-PAT-NO: 6314559
DOCUMENT-IDENTIFIER: US 6314559 B1
**** See image for Certificate of Correction ****

TITLE: Development system with methods for assisting a user with inputting source code

DATE-ISSUED: November 6, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Sollich; Peter Franz Valentin	Santa Cruz	CA		

US-CL-CURRENT: 717/111

ABSTRACT:

A visual development system having an interface which assists a user with input of source code expressions and statements during creation of a computer program is described. The interface includes an Integrated Development Environment (IDE) interface having a code editor with "Code Completion" and "Code Parameter" features for displaying context sensitive pop-up windows within a source code file. Code Completion is implemented at the user interface level by displaying a Code Completion dialog box after the user enters a record or class name followed by a period. For a class, the dialog lists the properties, methods and events appropriate to the class. For a record or structure, the dialog lists the data members of the record. To complete entry of the expression, the user need only select an item from the dialog list, whereupon the system automatically enters the selected item in the code. Code completion also operates during input of assignment statements. When the user enters an assignment statement for a variable and presses a hot key (e.g., <ctrl><space_bar>), a list of arguments valid for the variable is displayed. Here, the user can simply select an argument to be entered in the code. Similarly, the user can bring up a list of arguments when typing a procedure, function, or method call and needs to add an argument. In this manner, the user can view the required arguments for a method as he or she enters a method, function, or procedure call.

5 Claims, 35 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 34

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments   KWIC | Draw Desc | Image

□ 10. Document ID: US 6292933 B1

L6: Entry 10 of 18 File: USPT Sep 18, 2001

US-PAT-NO: 6292933
DOCUMENT-IDENTIFIER: US 6292933 B1
**** See image for Certificate of Correction ****

TITLE: Method and apparatus in a data processing system for systematically serializing complex data structures

DATE-ISSUED: September 18, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bahrs; Peter C.	Austin	TX		
Chancey; Raphael Poole	Austin	TX		
Feigenbaum; Barry Alan	Austin	TX		
Modh; Manish Mahesh	Round Rock	TX		
Sundberg; Sean Michael	Cedar Park	TX		
Woolfrey; John Allen Hubert	Mississauga			CA

US-CL-CURRENT: 717/107; 707/203, 717/108, 717/109

ABSTRACT:

A method and apparatus in a data processing system for serialization data. A serializer receives a data element for serialization, wherein the data element includes a class name string. Responsive to receiving the data element, the serializer replaces the class name string with a code having a smaller size than the class name string to form a modified data element. Responsive to forming the modified data element, in which the serializer serializes the modified data element. The serialized data is transmitted and deserialized by deserializer which replaces the indicator with the class name.

24 Claims, 197 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 119

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments   KWIC | Drawn Desc | Image

☐ 11. Document ID: US 6247020 B1

L6: Entry 11 of 18   File: USPT   Jun 12, 2001

US-PAT-NO: 6247020
DOCUMENT-IDENTIFIER: US 6247020 B1

TITLE: Development system with application browser user interface

DATE-ISSUED: June 12, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Minard; Jayson R.	Littleton	CO		

US-CL-CURRENT: 707/104.1; 345/629, 345/854, 709/206, 715/501.1, 715/513, 717/100, 717/107, 717/118

ABSTRACT:

A component-based, rapid application development (RAD) Java environment providing an improved user interface is described. The interface includes a single Application Browser or "AppBrowser" that is used to perform all the usual development functions. The AppBrowser lets the user explore, edit, design, and debug all in one unified window. Serving as a mechanism for hosting arbitrary documents or other objects related to development, the AppBrowser presents the documents or other objects for manipulation in a window that consists of three panes: Navigation pane, Content pane, and Structure pane. In general, the Navigation pane displays a list of documents, the Content pane displays the document itself, and the Structure pane displays the structure of the document if available. The grouping of the three panes exists on a browser "context" or mode. Multiple contexts can appear on an AppBrowser represented by a tab in a tab set at the lower left of the window. These contexts add a 3D feel to AppBrowser by layering its functionality in one window rather than spread across multiple windows. Switching between completely different contexts is then as easy as selecting a tab. The AppBrowser may have different logic for any

given context that is global across the three panes; the logic is provided by implementing a standard browser context interface.

33 Claims, 18 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 16

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments

KWIC | Draw Desc | Image

□ 12. Document ID: US 6237135 B1

L6: Entry 12 of 18 File: USPT May 22, 2001

US-PAT-NO: 6237135
DOCUMENT-IDENTIFIER: US 6237135 B1

TITLE: Development system with visual design tools for creating and maintaining Java Beans components

DATE-ISSUED: May 22, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Timbol; Michael	Scotts Valley	CA		

US-CL-CURRENT: 717/107; 717/118

ABSTRACT:

A component-based, rapid application development (RAD) Java system providing visual designers (i.e., wizards) and methodology allowing a developer (user) to create Java Beans-compatible components rapidly and easily is described. In typical operation of the system, the user may generate a "bean" component by invoking a wizard-based interface that implements methodology for automatically generating and managing the bean. The user employs the wizard to specify information about the bean, such as the name of the bean, the package it will be in, and the class it extends from. In response to the user input, the system creates a bean with the name the user specified, places it in the user's current project, and displays the source code generated for the bean. The user may visually interact with the bean (or other existing bean) by using the visual designers to manage the bean's properties, including: adding properties to the bean, adding bound and constrained properties, modifying properties of the bean, removing properties from the bean, or even generate a custom property editor. In a similar manner, the user may proceed to use the visual designers to manage the bean's events, including: adding events, listening for events, and creating custom event sets.

38 Claims, 15 Drawing figures
Exemplary Claim Number: 29
Number of Drawing Sheets: 13

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments

KWIC | Draw Desc | Image

□ 13. Document ID: US 6185728 B1

L6: Entry 13 of 18 File: USPT Feb 6, 2001

US-PAT-NO: 6185728
DOCUMENT-IDENTIFIER: US 6185728 B1

TITLE: Development system with methods for type-safe delegation of object events to event handlers of other objects

DATE-ISSUED: February 6, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Hejlsberg; Anders	Los Altos Hills	CA		

US-CL-CURRENT: 717/109; 717/116

ABSTRACT:

A visual development system is described which provide "method pointers" allowing a developer/user to achieve delegation between objects programmatically as well as visually. Delegation "binds" an event source with an event handler. This binding is directly supported within the programming language of the system. Using a simple assignment statement, the user can assign a handler of one object (i.e., delegated-to object) to the method pointer of another object (i.e., delegated-from object). A delegation through assignment leads to not only the address of the method being copied but also the address of the delegated-to object (i.e., "this" pointer) being copied into the method pointer. The method pointer, when called, can invoke the pointed-to method on that (delegated-to) object, complete with the context (e.g., local data) of the object. The object knows about itself and, thus, can provide the appropriate context. In this fashion, the system allows the user to easily control object behavior through delegation, whether the user is working in a visual environment or a programming environment (or switching back and forth between each).

35 Claims, 20 Drawing figures
Exemplary Claim Number: 21
Number of Drawing Sheets: 18

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments   KWIC | Draw Desc | Image

□ 14. Document ID: US 5987247 A

L6: Entry 14 of 18 File: USPT Nov 16, 1999

US-PAT-NO: 5987247
DOCUMENT-IDENTIFIER: US 5987247 A
**** See image for Certificate of Correction ****

TITLE: Systems, methods and computer program products for building frameworks in an object oriented environment

DATE-ISSUED: November 16, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Lau; Christina	Ontario			CA

US-CL-CURRENT: 717/100; 717/108

ABSTRACT:

Systems, methods and computer program products enable framework building by interactively generating a framework corresponding to a design and editing the interactively generated framework to obtain a completed framework. The framework is displayed in various stages of construction or modification using a number of different views including a tree view, a graph view, a method view and an edit view. In addition, parts of the framework are identified as requiring completion, as completed, and as permitting completion. Still further, user methods and framework methods are differentiated. Help guided actions are provided and may be invoked to assist in the development process. Once the framework has been completed, it can be stored in a data model, and code can be generated from the completed framework. Finally, interfaces developed using conventional editors may be parsed and imported into the data model for later code generation. As a result of the present invention, code may be generated based on the completed framework and executed in a distributed computing environment.

76 Claims, 6 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments   KWIC | Draw Desc | Image

□ 15. Document ID: US 5724589 A

L6: Entry 15 of 18   File: USPT   Mar 3, 1998

US-PAT-NO: 5724589
DOCUMENT-IDENTIFIER: US 5724589 A

TITLE: Development system with a property-method-event programming model for developing context-free reusable software components

DATE-ISSUED: March 3, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Wold; Ivar	Boston	MA		

US-CL-CURRENT: 709/318; 717/100, 717/116, 717/140

ABSTRACT:

A development system providing a property-method-event programming (PME) model for developing context-free reusable software components is described. Despite the absence of any C++ language support for events, the present invention provides a type-safe "wiring" mechanism--one using standard C++ to dispatch an event, raised by one object (the "event source"), to a method of another object (the "event sink"), with the requirement that the event source does not need to know the class of the event sink. As a result, the system allows developers to create C++ software components which can be connected together without the components having to know anything about the makeup of the component to which it is connected. Thus, developers can create pre-packaged, re-usable software components which can simply be "plugged into" a design--all accomplished within the confines of the standard C++ programming language (i.e., without having to employ proprietary extensions).

30 Claims, 10 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 6

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments   KWIC | Draw Desc | Image

□ 16. Document ID: US 5630131 A

L6: Entry 16 of 18   File: USPT   May 13, 1997

US-PAT-NO: 5630131
DOCUMENT-IDENTIFIER: US 5630131 A

TITLE: Method and apparatus for importing and exporting archive files for a graphical user interface

DATE-ISSUED: May 13, 1997

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Palevich; Jack H.	Sunnyvale	CA		
Taylor; Thomas H.	San Jose	CA		

US-CL-CURRENT: 717/108; 345/700, 709/315, 717/109

ABSTRACT:

A method for use in an object oriented programming (OOP) based computer system which utilizes a first application program having interface objects for building a second application program having additional interface objects is provided. The method includes storing objects in a hierarchical locale tree in a storage device. The locale tree has a root locale level and at least one other locale level associated with the second locale. A first plurality of user interface objects having one or more interface object control and a second plurality of user interface objects not having interface object controls of the first locale are created. The second plurality of user interface objects are stored in the other locale level associated with the second locale. The locale tree is traversed, starting at the other locale level associated with the second local and proceeding to the root locale level, to assemble a set of available user interface objects from the user interface objects stored in the other locale level and the root locale level. User interface objects are utilized to develop a third application program derived from the first and second application programs. In addition, objects are imported and exported during processing for use in other program applications by segmenting an object into a human readable and a binary representation. In addition, a storage device readable by a computer system for implementing the method and a user interface object archive system which implements the method are provided.

15 Claims, 39 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 36

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments   KWIC | Drawn Desc | Image

---

☐ 17. Document ID: US 5519862 A

L6: Entry 17 of 18   File: USPT   May 21, 1996

US-PAT-NO: 5519862
DOCUMENT-IDENTIFIER: US 5519862 A

TITLE: Concurrent processing apparatus with incremental command objects

DATE-ISSUED: May 21, 1996

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Schaeffer; Arnold	Belmont	CA		
Goldsmith; David B.	Los Gatos	CA		
Moeller; Christopher P.	Los Altos	CA		
Heninger; Andrew G.	Mountain View	CA		

US-CL-CURRENT: 717/165; 717/110

ABSTRACT:

A method and apparatus for an innovative object oriented framework system is

disclosed. The system uses an innovative load architecture for a framework application by multiple users. The load architecture implements functions, static data and classes in a more flexible manner than prior operating systems.

42 Claims, 30 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 19

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments   KWIC | Draw Desc | Image

☐ 18. Document ID: US 5479601 A

L6: Entry 18 of 18   File: USPT   Dec 26, 1995

US-PAT-NO: 5479601
DOCUMENT-IDENTIFIER: US 5479601 A

TITLE: Method and apparatus for processing commands generated by user interface controls in an atomic manner

DATE-ISSUED: December 26, 1995

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Matheny; John R.	Mountain View	CA		
White; Christopher	Mountain View	CA		
Anderson; David R.	Cupertino	CA		

US-CL-CURRENT: 345/700; 345/701, 717/168

ABSTRACT:

An object-oriented user interface utilizes object-oriented controls that operate together as a single, atomic group to change data values and are affected as a group by conventional editing "undo" and "redo" actions. In accordance with one embodiment, each control in the group generates a command which modifies a stored control value when the control is manipulated by a user. In response to user activation, a group acceptance control generates a command which causes the data values to be changed to the stored control values. In accordance with another embodiment, each control in the group generates a command which modifies the group acceptance control command. When the group acceptance control command is finally activated, the modified command causes the data values to be changed. The entire control group can also be undone and redone in a single atomic operation which is implemented by placing a mark on an undo stack when an interface session involving a control group is started. When the session ends, all of the commands executed since the mark was placed on the undo stack are collected together into a single command group which can be undone or redone as a unit.

16 Claims, 20 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 13

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments   KWIC | Draw Desc | Image

Generate Collection | Print

Terms	Documents
L5 AND ((717/$$$)!.CCLS.)	18

Display Format: REV Change Format

Previous Page Next Page